Please amend the claims as follows:

1. (Currently Amended) A method of executing a ~~transaction~~ task within a transaction processing system, the method including:

responsive to an event, identifying a workflow associated with the event;

distributing ~~a~~ the task, which at least partially executes the workflow, from a task queue to an available thread within a pool of threads operating within a multiprocessor system, the distributing of the task to the available thread being responsive to dynamically assigning a new priority to the task;

identifying a processor affinity attributed to the task; and

assigning the available thread to a processor within the multiprocessor system according to the processor affinity attributed to the task,

wherein the event comprises a transaction event and the task comprises a transaction routing task, responsive to a transaction request associated with the transaction event, that routes the transaction request to an agent of the transaction processing system.

2. (Cancelled)

3. (Cancelled)

4. (Cancelled)

5. (Original) The method of claim 1 wherein the task has a real-time priority and is distributed in accordance with the real-time priority to the available thread within the pool of threads.

6. (Canceled)

7. (Original) The method of claim 1 including assigning the available thread to a processor within the multiprocessor system according to a thread priority.

8. (Original) The method of claim 7 including assigning the thread priority to the available thread based on a priority of the task distributed to the available thread.

9. (Currently Amended) Apparatus for executing a ~~transaction~~ task within a transaction processing system, the apparatus comprising:

a dispatcher to identify a workflow associated with an event;

a scheduler to issue ~~a~~ the task that at least partially executes the workflow associated with the event, the scheduler to issue the task from a task queue, responsive to a dynamic assignment of a new priority to the task; and

a thread within a pool of threads operating within a multiprocessor system to execute the task, the dispatcher to identify a processor affinity attributed to the task, and to assign the thread to a processor within the multiprocessor system according to the processor affinity attributed to the task, wherein the event comprises a transaction event and the task comprises a transaction routing task, responsive to a transaction request associated with the transaction event, that routes the transaction request to an agent of the transaction processing system.

10. (Previously Presented) The apparatus of claim 9 wherein the dispatcher is to generate the task that at least partially executes the workflow.

11. (Previously Presented) The apparatus of claim 10 including a task queue to which the task is dispatched by the dispatcher, and from which the thread within the pool of threads receives the task.

12. (Previously Presented) The apparatus of claim 11 including a scheduler that is to issue the task from the task queue to the thread within the pool of threads.

13. (Previously Presented) The apparatus of claim 12 wherein the scheduler is to issue the task from the task queue to the thread within the pool of threads based on the priority associated with the task.

14. (Canceled)

15. (Previously Presented) The apparatus of claim 13 wherein the scheduler is to issue the task from the task queue according to a real-time priority assigned to the task.

16. – 18. (Canceled)

19. (Previously Presented) The apparatus of claim 9, wherein the scheduler is to assign the thread to a processor within the multiprocessor system according to a thread priority.

20. (Previously Presented) The apparatus of claim 19, wherein the scheduler is to assign the

thread priority to the thread based on a priority of the task distributed to the thread.

21 -23 (Canceled)

24. (Previously Presented) The method of claim 1, further including determining a best match between the task and the available thread.

25. (Previously Presented) The method of claim 1, wherein the available thread is a member of a class of threads that are included in the pool of threads, the class of threads being associated with the priority.

26. (Previously Presented) The apparatus of claim 9, wherein the scheduler is to determine a best match between the task and the available thread

27. (Previously Presented) The apparatus of claim 9, wherein the available thread is a member of a class of threads that are included in the pool of threads, the class of threads being associated with the priority.